Do Humans Look Where Deep Convolutional Neural Networks "Attend"?

Mohammad K. Ebrahimpour[1], J. Ben Falandays[2], Samuel Spevack[2], and David C. Noelle[1,2]

[1] EECS, University of California, Merced
{mebrahimpour,dnoelle}@ucmerced.edu
[2] Cognitive & Information Sciences, University of California, Merced
{bfalandays,sspevack}@ucmerced.edu

Abstract. Deep Convolutional Neural Networks (CNNs) have recently begun to exhibit human level performance on some visual perception tasks. Performance remains relatively poor, however, on some vision tasks, such as *object detection*: specifying the location and object class for all objects in a still image. We hypothesized that this gap in performance may be largely due to the fact that humans exhibit *selective attention*, while most object detection CNNs have no corresponding mechanism. In examining this question, we investigated some well-known attention mechanisms in the deep learning literature, identifying their weaknesses and leading us to propose a novel attention algorithm called the*Densely Connected Attention Model*. We then measured human spatial attention, in the form of eye tracking data, during the performance of an analogous object detection task. By comparing the learned representations produced by various CNN architectures with that exhibited by human viewers, we identified some relative strengths and weaknesses of the examined computational attention mechanisms. Some CNNs produced attentional patterns somewhat similar to those of humans. Others focused processing on objects in the foreground. Still other CNN attentional mechanisms produced usefully interpretable internal representations. The resulting comparisons provide insights into the relationship between CNN attention algorithms and the human visual system.

Keywords: Visual Spatial Attention· Computer Vision · Convolutional Neural Networks · Densely Connected Attention Maps · Class Activation Maps · Sensitivity Analysis

1 Introduction

Recent years have seen huge advances in the use of artificial neural networks (ANNs) in a wide variety of machine learning applications. One particularly promising domain is computer vision, for which deep Convolutional Neural Networks (CNNs) have been proven successful in tasks such as classification, semantic segmentation, image captioning, and object detection. The impressive capabilities of such networks, which have met or even surpassed human performance in some vision tasks [11], are somewhat surprising given how little is actually understood about their internal operations. Due to the large number of nonlinear interactions inside these networks, ANNs have long been treated as "black boxes", with their inner workings opaque to even their creators.

However, just as the “black box” perspective on the human mind gave way to the development of new methods and theoretical tools, researchers in computer science have recently begun finding new ways of understanding the intermediate representations produced by ANNs [21]. For example, methods of “deep visualization” examine the representations learned by individual artificial neurons by iteratively generating a synthetic input image that maximally activates each neuron. The images produced in this process are often surprisingly interpretable by the human eye [11], suggesting that these networks may learn feature representations that are perhaps similar to those of the human visual system.

Of course, ANNs have have been at the heart of powerful models of human cognitive processes for over three decades [15]. In the field of computational cognitive neuroscience, ANNs have served a crucial role in discriminating between various proposed models of the structure and dynamics of cognitive and brain systems [12]. By comparing the performance of human participants to ANNs of various designs, researchers can investigate the feasibility of specific network models for explaining aspects of brain and behavior. For example, while most CNNs designed for object recognition are purely feedforward, Rajaei and colleagues recently argued, based on neuroimaging data and computational models, that recurrent connections are crucial for performance under cases of degraded input, such as partial object occlusion [13]. This serves to illustrate that, by examining where humans succeed and ANNs fail, we can improve both our understanding of the brain and create more effective computer vision systems.

In this spirit, computer scientists have been taking inspiration from the human visual system to improve the speed and accuracy of CNNs for object detection. In particular, visual selective attention has been proposed as one mechanism that is crucial for human object detection performance, but such a mechanism is absent from most algorithms designed for the same purpose. Several CNN techniques have now been proposed that implement some form of selective attention [23, 16, 4], but it is unknown how these attention algorithms compare to human selective attention. Examining the differences and similarities between human overt visual attention and the attentional representations produced in these CNNs will help to hone our understanding of specifically which features of the human visual system may most fruitfully be modeled by CNNs (and for which tasks), or, indeed, whether deep networks ofter a promising approach to understanding human vision, at all. Given the relative power of human vision in comparison to state-of-the-art object detection CNNs, comparisons of this kind might also suggest new and better biologically-inspired approaches to selective attention in computer vision.

For the sake of comparison, we took human behavior to be approximately normative with regard to the allocation of attention. We recorded the eye motions of humans as they detected objects in still images, and we took these data as indicating the image regions most worthy of overt selective attention. We then examined a variety of CNN approaches to attention, including a new architecture, proposed here, called the *Densely Connected Attention Model*. We assessed the approaches with regard to their correspondance to human performance. The resulting analyses have produced insights into both human selective attention and the design of computer vision object detection systems. By identifying the layers within the Densely Connected Attention Model that

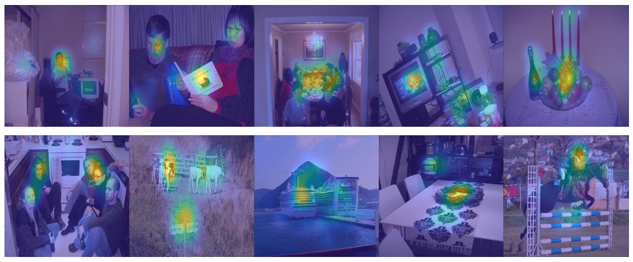

Fig. 1. Results of the human eye tracking study on some of our test images. The heatmaps reveal the distribution of attention.

best capture human selective attention, we were able to identify the sort of visual features that best predict the distribution of human eye fixations. By comparing various CNN methods, we were able to characterize their relative strengths and weaknesses for attention-guided object detection.

2 Eye Tracking Study

Participants. Our participants were 15 healthy undergraduate students (12 female, 3 male; age: mean±s.d.= 20.06 ±1.62) at University of California, Merced. Participants received one hour of course credit for their participation. Participants provided informed consent in accordance with IRB protocols. Participation was restricted to individuals with normal vision, as reported on a pre-screen survey.
Materials. The stimuli were a subset of 200 images drawn from the PASCAL Visual Object Classes 2007 database [6]. Each image was scaled up to twice its original size for display on a 1920x1080 screen. The display width of images was between 636-1000 pixels, subtending 23.83-34.76 degrees of visual angle. The display height of images was between 350-1000 pixels, subtending 13.68-34.76 degrees of visual angle. Images were centered on a black background.
Procedure. Participants completed an object detection task individually in the lab. Participants were seated at a desk in front of a computer screen and wore a head-mounted Eyelink II eye tracker. The eye tracker was set to pupil/corneal reflection recording mode and sampled at a rate of 250 Hz. A microphone was placed near participants to record their speech. Before beginning the task, the eye tracker was calibrated with a nine-point grid. Participants were also shown how to perform a drift correction, which occurred prior to each trial. Eye movement data was collected with the Eyelink software and custom Matlab scripts, implemented with the Psychophysics toolbox Matlab package [2].

Each trial started with a drift correction in which participants had to fixate a center dot and press the space bar to initiate the trial. Participants were allowed to pause on the drift correction screen for as long as they wished before initiating a trial. Then, a randomly selected image was displayed on the screen for 5000 ms. Participants were instructed to name out loud as many unique objects as they could detect within the time limit. This was repeated until participants had viewed all 200 images, once each. The

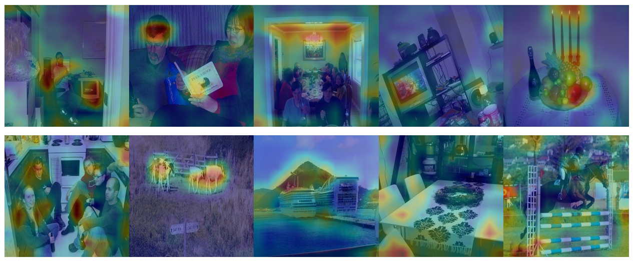

Fig. 2. Results of the CAM attention algorithm. The heatmaps reveal the attention of the CNN.

experiment took about 30 minutes to complete, with $\sim$10 minutes dedicated to setup and calibration and $\sim$20 spent on the task itself.

Data Processing. The raw eye tracking data was first converted into a Matlab-compatible data structure using the `Edf2Mat` package. Then, custom scripts were used to generate fixation heatmaps for each image. Recorded fixations contributed to the heatmaps only if they fell entirely within the period of display. Fixations were first pooled from all participants. Then, for each image a zero matrix was generated with the same dimensions as the pixel dimensions of the original (before scaling for display) image. Fixation coordinates from the display images were then scaled down to map onto the coordinates of the original sizes. Each possible fixation position then corresponded to a position within the zero matrix for that image. For each recorded fixation at a given location, the value of the corresponding cell in the zero matrix was increased by the duration of the fixation in milliseconds. Then, all values were divided by the maximum value in the matrix such that possible values ranged between zero and one. Finally, we performed a convolution over the matrix with a Gaussian kernel ($\sigma = 20$, size = 80×80). This process generated a fixation heatmap showing the relative likelihood of fixations occurring at each region of each image. Examples of the human eye tracking overt visual attention maps are illustrated in Figure 1.

3 Attention Models

In recent years, deep CNNs have demonstrated human level performance on image classification tasks when given large amounts of supervised training data. This impressive performance has led to a growing interest in understanding the internal representations learned by the networks [5, 22, 21, 17]. Of particular interest, here, are the mechanisms of "selective attention" in these networks that allow them to focus on relevant portions of input images. In this section, we review several previously published approaches and then propose a novel CNN architecture: the *Densely Connected Attention Model*.

3.1 Class Activation Maps

Zhou et al. proposed a technique called *Class Activation Maps (CAM)* [23]. These were predicated on the observation that deep CNN architectures for computer vision lose spatial information close to the network output due to the use of fully connected layers after the convolutional layers. The authors proposed "chopping off" the fully connected layers and calculating the *Global Average Pooling (GAP)* of activation at the last convolutional layer, aggregating the spatial information in each channel. For a given image, let $f_k(x, y)$ represent the activation of filter k in the last convolutional layer at spatial location (x, y). Then, for filter k, the result of performing GAP is $F^k = \sum_{x,y} f_k(x, y)$. A linear regression can be conducted between the F^k values and each output score, S_c, for each object class, c, producing regression weights, w_k^c. Given these weights, the CAM for a given image, with regard to object class, c, is:

$$M_c(x, y) = \sum_k w_k^c f_k(x, y) \tag{1}$$

The authors argued that using this approach would preserve more spatial location information for the main objects in the image. Examples of CAMs are depicted in Figure 2. This method has a number of drawbacks. Specifically, determining the regression weights takes time, and using the attention information in the suggested manner can noticeably decrease classification performance.

Gradient Based Class Activation Maps. With the goal of improving on the CAM method, Selvaraju and colleagues proposed a different approach [16]. They argued that the weights for each channel that are used in CAMs are implicit in the CNN, itself, so there is no need to perform additional regressions [16]. They suggested taking the derivative of the winning (i.e., greatest) object class output with regard to the activation in the last convolutional layer of the network:

$$A = \frac{\partial y_c}{\partial f} \tag{2}$$

where y_c is the output activation for object class, c, and $f \in \mathbb{R}^{W \times H \times K}$ is the last convolutional layer. The derivative values are aggregated channel-wise into a matrix, A, and the weights for each channel are computed as:

$$a_k^c = \frac{1}{W \times H} \sum_{x \in W} \sum_{y \in H} A(x, y) \tag{3}$$

This weight, a_k^c, is intended to capture the “importance” of channel k for a target class c. The corresponding attention maps are calculated similarly to CAMs, though values are rectified (ReLU) to be positive:

$$L^c_{Grad-CAM} = ReLU(\sum_k a_k^c f_k) \tag{4}$$

Notice that this results in a coarse heatmap of the same size as the convolutional feature maps (14×14 in the case of the last convolutional layers of the VGG network). Since this method made use of gradient information, it is called *Grad-CAM*. Example Grad-CAM attention maps are illustrated in Figure 3.

Guided-Backpropagation. The *Guided-Backpropagation* technique was proposed as a way to train CNNs so as to make activation in convolutional feature maps easier to interpret [19]. The goal was to encourage units to only become active if they are contributing to the activation of the appropriate object class output. This would make the

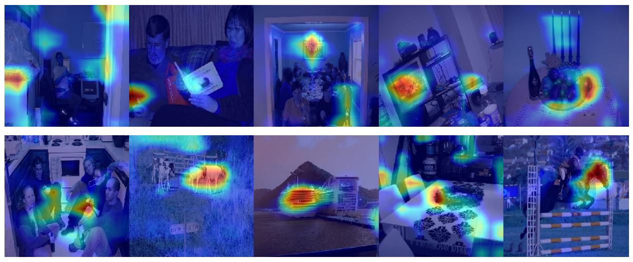
Fig. 3. Results of the Grad-CAM attention algorithm. The heatmaps show attended regions.

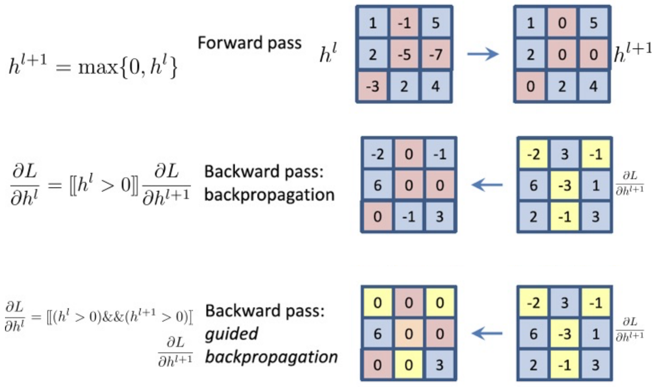
$h^{l+1} = \max\{0, h^l\}$
Forward pass
h^l
h^{l+1}
$\frac{\partial L}{\partial h^l} = [[h^l > 0]] \frac{\partial L}{\partial h^{l+1}}$
Backward pass: backpropagation
$\frac{\partial L}{\partial h^{l+1}}$
$\frac{\partial L}{\partial h^l} = [[(h^l > 0)\&\&(h^{l+1} > 0)[[\frac{\partial L}{\partial h^{l+1}}$
Backward pass: guided backpropagation

Fig. 4. An illustration of the training process used in guided-backpropagation.

spatial location of active units indicative of where objects were. The method involved modifying the standard *backpropagation of error* algorithm [15] by surpressing negative gradients. Unit error (delta) values were rectified, hindering the formation of strong negative connection weights. The bias toward positive weights made the "meaning" of learned features easier to interpret. A very simple example of this learning process is illustrated in Figure 4. Figure 5 shows the resulting attention maps (where activation is high) for some example images.

Guided-Gradient Based Class Activation Maps. In order to obtain information about both relevant image regions and relevant pixels within regions, Selvaraju and colleagues fused Guided-Backpropagation with Gradient Based Class Activation Maps, producing the *Guided Grad-CAM* method [16]. The merging of the previous algorithms was done in a simple manner. The attention maps resulting from the two algorithms were combined using point-wise multiplication. Some example results are displayed in Figure 6.

The reviewed attention algorithms so far are using the last convolutional layer as their main source for attention. The encapsulated information in the last convolution layer is mostly abstract and it has lost lots of spatial information due to the pooling under sampling layers. To overcome this issue, we suggest using the hidden information in all layers, the early ones as well as the deepest ones, to maximize the sophisticated information as well as the spatial information of the input image.

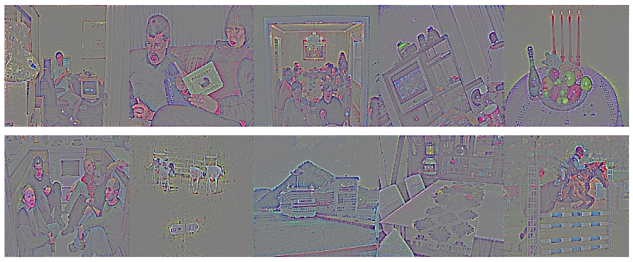

Fig. 5. Results of the Guided-Backpropagation attention algorithm. The color-coded maps show attended pixels.

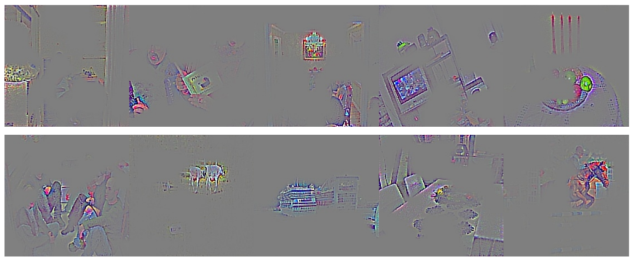

Fig. 6. Results of the Guided-Grad-CAM attention algorithm. The color-coded maps show attended pixels.

3.2 The Densely Connected Attention Model

All of these CNN attentional mechanisms rely on activation in the last convolutional layer. Information in earlier layers of the networks is ignored. It is possible, however, that useful guidance for attention might be found in the earlier convolutional layers, where spatial resolution tends to be higher and detected visual features tend to be more simple and smaller. Our proposed *Densely Connected Attention Model* assembles information from each channel and each spatial location across all of the layers in the CNN image classifier. Our method is fast and efficient, and it does not require any additional training of the CNN. A schematic of our approach is shown in Figure 7.

Spatial Attention. Generally speaking, objects occupy only portions of images, leaving background regions that can distract and misinform object detection systems. Instead of considering all parts of an image equally, spatial attention can focus processing on foreground regions, supporting the extraction of features most relevant for determining object class and object extent.

Formally, the annotation that we use to represent the activation of convolutional feature layer n is $f_n \in \mathbb{R}^{W \times H \times C}$, where W and H are the spatial dimensions of the rectangular layer and C is the number of feature channels in the layer. Spatial positions are specified by coordinate pairs: $\mathbb{L} = \{(x, y) | x = 1, 2, \ldots, W; y = 1, 2, \ldots, H\}$.

For layer n in a pretrained image classification network, the layer-specific spatial attention map is . . .

$$A_n^s = W_n^s \odot f_n \tag{5}$$

...where W_n^s are weights that indicate the importance of each spatial location, across all of the convolutional channels. We initially calculate these weights based on the sensitivity of the *Gestalt Total (GT)* activation of the network to the feature [4]. The Gestalt Total is calculated from the activation of the last convolutional layer, A_{last}, as follows:

$$GT = \frac{1}{H \times W \times C} \sum_{i,j,k} A_{last}^s(i, j, k) \tag{6}$$

The sensitivity of GT to a feature at layer n is ...

$$G_n = \frac{\partial GT}{\partial f_n} \tag{7}$$

$$\hat{W}_n^s(x, y) = \sum_c^C G_n(x, y, c), \tag{8}$$

...where $\hat{W}_n^s$ is not normalized (i.e., the weights are not in the $[0, 1]$ range). To normalize the weights for each location, we apply a softmax operation to the weights spatially ...

$$W_n^s(x, y) = \frac{\exp(\hat{W}_n^s(x, y))}{\sum_{i \in W, j \in H)} \exp(\hat{W}_n^s(i, j))}, \tag{9}$$

...where $W_n^s(x, y)$ denotes the weight for location (x, y) in layer n.

Channel-Wise Attention. The spatial attention calculation assigns weights to spatial locations, which addresses the problem of distractions from background regions. There is another way in which distractions can arise, however. Specific channels at a given layer can be distracting. When dealing with convolutional features, most of the existing methods treat all channels without distinction. However, different channels often have different degrees of relevance for objects of specific classes. Here, we introduce a channel-wise attention mechanism that assigns larger weights to channels for which the GT is sensitive, given the currently presented image. Incorporating these channel-wise attentional weights are intended to reduce this kind of distracting interference.

For channel-wise attention, we unfold f_n as$f = [f_n^1, f_n^2, \ldots, f_n^C]$, where $f_n^i \in \mathbb{R}^{W \times H}$ is the ith channel slice of f_n, and C is the total number of channels. The goal is to calculate a weight, W_n^c, to scale features according to a channel-specific assessment of relevance, allowing for the construction of a layer-specific channel-wise attention map:

$$A_n^c = W_n^c \cdot f_n \tag{10}$$

Computing W_n^c is facilitated by the fact that we already have the sensitivities, G_n. Thus, an initial value for the weights can be found by setting $\hat{W}_n^c(c) = \sum_{x \in W, y \in H} G_n(x, y, c)$. These weights can be normalized to the $[0, 1]$ range using the softmax function:

$$W_n^c(c) = \frac{\exp(\hat{W}_n^c(c))}{\sum_{i \in C} \exp(\hat{W}_n^c(i))}, \tag{11}$$

These are the final channel-wise weights for layer n.

Dense Attention Maps. Given the spatial attention weights and the channel-wise attention weights, an attention weighted feature for layer n is calculated as ...

$$f_n^{SCA} = A_n^c \cdot f_n + A_n^s \odot f_n, \tag{12}$$

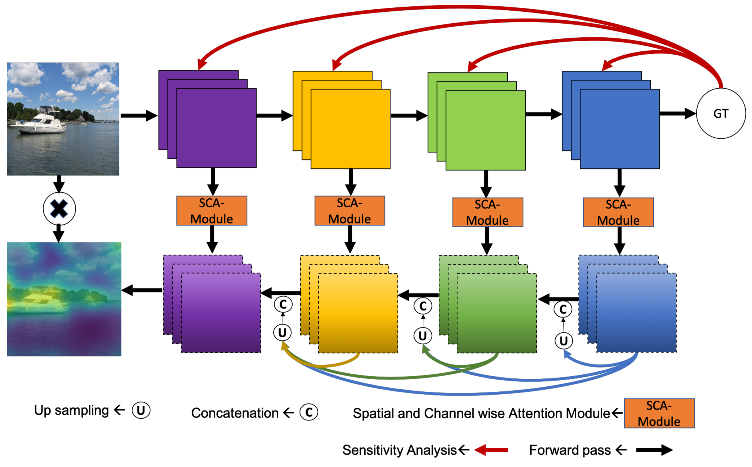

Fig. 7. The network architecture of the *Densely Connected Attention Model*.

. . . with $f_n^{SCA} \in \mathbb{R}^{W\times H\times C}$. (In Figure 7, this computation is done in the *Spatial and Channel wise Attention Module – SCA-Module*.) For the last convolutional layer, m, the attention map is simply the weighted features: $A_m = f_m^{SCA}$. For earlier layers, the weighted features are concatenated across layers, incrementally from the last layer to the first, but this is done after up-scaling lower spatial resolution (later) convolutional layers:

$$A_i^{SCA} = [UP(f_m^{SCA}), UP(f_{m-1}^{SCA}), \dots, f_i^{SCA}] \\ i = \{1, \dots, m-1\} \tag{13}$$

This process generates dense combination maps that are intended to incorporate both semantic information from the late layers and higher resolution spatial information from the early layers. The maps are aggregated to produce a single attention map for the whole image. Since each layer can have a different number of channels, we simplify this aggregation by averaging each layer’s attention map across channels, transforming each A_i^{SCA} into a $W \times H$ matrix. The aggregated attention maps at each layer are then computed as:

$$A_i = A_m^{SCA} + A_{m-1}^{SCA} + \ldots + A_i^{SCA}. \tag{14}$$

It is important to note that the attention weights (W_n^s and W_n^c) are not learned as part of a training process. We begin with a pretained image classification network (VGG16 [18]), and the attention weights are efficiently calculated for each layer when a given image is presented to that network. Some resulting attention maps from our model are shown in Figure 8.

4 Comparing CNN and Human Attention Maps

We have reviewed several existing CNN attention algorithms, and we have proposed a novel Densely Connected Attention Model which incorporates spatial attention as

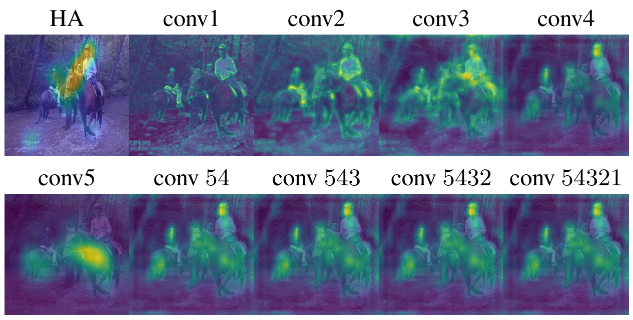

Fig. 8. Results of the Densely Connected Attention Model for each convolutional layer, as well as for combinations of layers. The numbers following the "conv" label indicate layer number. For instance, "conv 54" refers to the combination of layers 5 and 4. Human eye tracking results for this example image are also shown (HA).

well as channel-wise attention in each layer. We compared the attention maps of the various CNNs with overt visual attention maps produced from our eye tracking data, using exactly the same images in all cases. Our summary comparison statistic was the mean absolute error (MAE) between attention maps ...

$$MAE = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} |S(x,y) - G(x,y)| \tag{15}$$

... where W and H are the width and height of the image, $S(.)$ is the attention map from a network, and $G(.)$ is the "ground truth" (human performance) attention map. We compared different CNN attention methods, but we also examined the fit of various attention maps at individual internal layers of the Densely Connected Attention Model. The resulting error values are summarized in Figure 9.

These results reveal that the CAM algorithm provides the closest match to overt visual human attention on our test images, while the second layer in our proposed attention mechanism comes in second place. The fact that layer 2 of our network is much closer to the attentional patterns of human participants than later layers is quite surprising, given the total reliance on the final convolutional layer seen in other mechanisms. However, the combination of layers in our attention mechanism makes the attentional pattern somewhat dissimilar to that of humans.

5 Discussion

We investigated various deep CNN visual attention mechanisms and compared their attentional patterns to that of human participants. We also proposed a novel attention algorithm which integrates more kinds of information, at multiple scales, than the other approaches. We found that the CAM algorithm provides the best match to human performance, with the second convolutional layer of our Densely Connected Attention Model ranking second. In general, none of the CNN algorithms provided a paricularly good

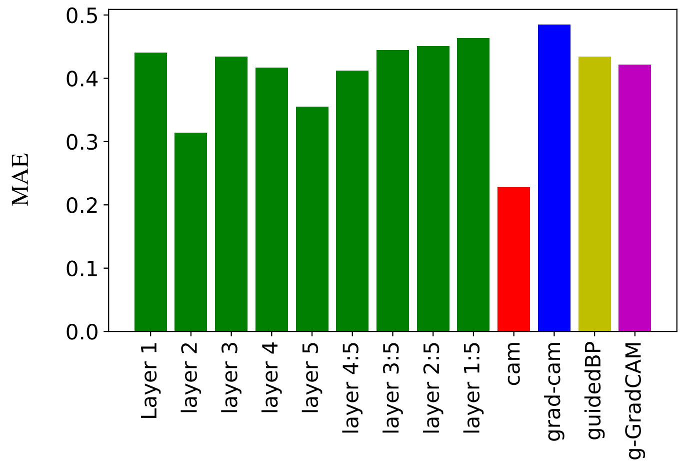

Fig. 9. Error between human attention maps and those produced at various layers of the Densely Connected Attention Model. Also shown are errors for other algorithms.

match to overt visual human attention, however. Thus, while deep CNNs may learn a hierarchy of visual features similar to the response properties of neurons in the human visual system [20], current attentional mechanisms in CNNs do not seem to align with human overt attention.

This suggests that human attention may not be a good guide for improving the object detection performance of deep CNNs. We have found that the attention maps produced by CAM, Grad-CAM, Guided-Backpropagation, and Guided Grad-CAM tend to focus on a single salient object in the image. In contrast, the Densely Connected Attention Model appears to attend to all objects in the image, ignoring background distractions. Despite these benefits, the CNN attention maps in our model were quite different than those of humans, with the greatest similarity appearing in layers that encode fairly low-level features. Interestingly, the results suggest that the distribution of human spatial attention is largely driven by low level visual features, as evidenced by the better performance of layer 2. We are left with interesting questions concerning the nature of the differences between CNN object detection and human vision that give rise to this mismatch of attentional patterns.

References

1. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR (2016)
2. Brainard, D.H., Vision, S.: The psychophysics toolbox. Spatial Vision **10**, 433–436 (1997)
3. Dai, K.J., R-FCN, Y.L.: Object detection via region-based fully convolutional networks. arxiv preprint. arXiv preprint arXiv:1605.06409 (2016)
4. Ebrahimpour, M.K., Li, J., Yu, Y.Y., Reese, J.L., Moghtaderi, A., Yang, M.H., Noelle, D.C.: Ventral-dorsal neural networks: Object detection via selective attention. In: WACV (2019)
5. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network. University of Montreal **1341**(3), 1 (2009)

6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision **88**(2), 303–338 (2010)
7. Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)
8. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: CVPR (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV (2016)
11. Nguyen, A., Yosinski, J., Clune, J.: Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616 (2016)
12. O'Reilly, R.C., Munakata, Y.: Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain. MIT Press (2000)
13. Rajaei, K., Mohsenzadeh, Y., Ebrahimpour, R., Khaligh-Razavi, S.M.: Beyond core object recognition: Recurrent processes account for object recognition under occlusion. bioRxiv p. 302034 (2018)
14. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS (2015)
15. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations. MIT Press (1986)
16. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
17. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
19. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806 (2014)
20. Yamins, D.L.K., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., DiCarlo, J.J.: Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences **111**(23), 8619–8624 (2014)
21. Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., Lipson, H.: Understanding neural networks through deep visualization. arXiv preprint arXiv:1506.06579 (2015)
22. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV (2014)
23. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR (2016)
24. Zhou, P., Ni, B., Geng, C., Hu, J., Xu, Y.: Scale-transferrable object detection. In: CVPR (2018)